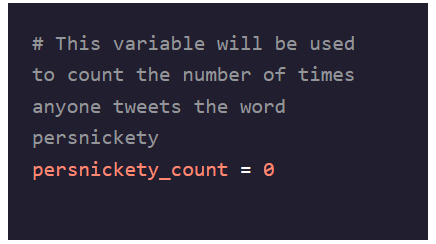
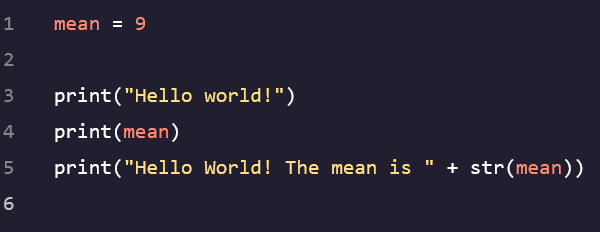
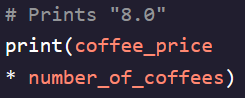
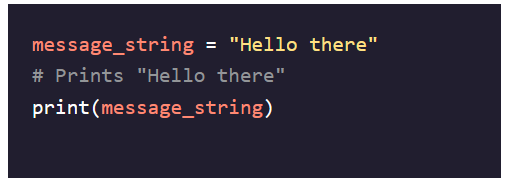
**Ignore Text:**

- Use # to tell program to ignore this bit of text   
- Used for documentation, providing code clarity on functions, or ignore a line of code for testing purposes  


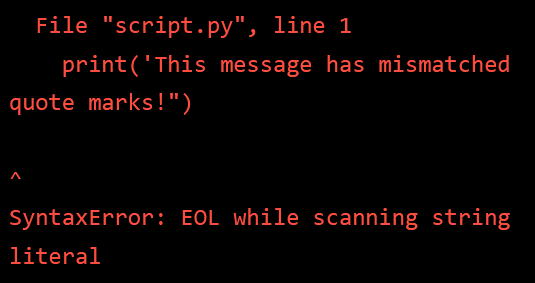
**Print:**

- Use print() to print out commands  
- Use print(“xxxxx”) to print out strings of words  
- Can append text with + to string together disparate parts  
- When converting a float/integer into a print statement with text, you **must** encapsulate it in a **str()** statement first  


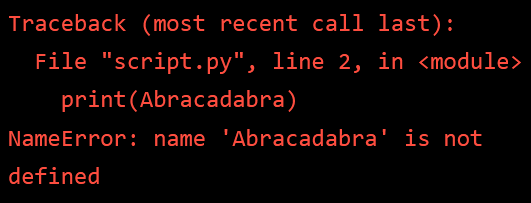
**Variables:**

- Used for storing information in an easy to retrieve, unique bucket  
- Use = to assign a value to a variable  
- Use \_ to breakup long variable names into easier to read chunks  
- Cannot begin with numbers but can contain numbers after the first letter  
- Python DOES NOT KNOW what words are, abcdx is just as valid a variable name as student\_ids  
- Can perform arithmetic operations on variables with no problem, doesn’t change their values  


**Syntax Errors:**

- Something is wrong with the way your program is written (incorrect punctuation, unexpected command, missing parenthesis, etc…breaks the design of Python)  


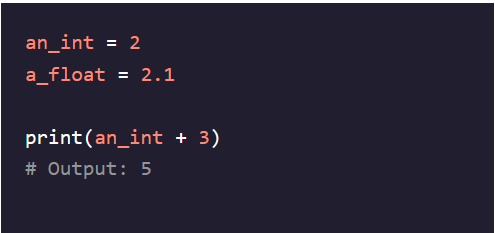
**Name Errors:**

- Occurs when Python sees a word that it does not recognize (undefined variables)  


**Integers (int):**

- Contains all counting numbers (1, 2, 3, etc..), their negative counterparts (-1, -2, -3, etc…) and

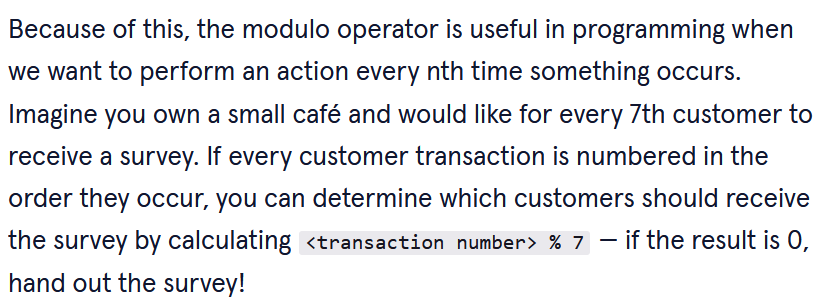
**Floating Point Numbers (float):**

**-** Decimal numbers and fractional quantities   


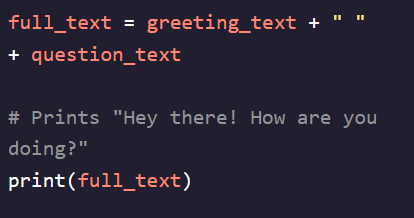
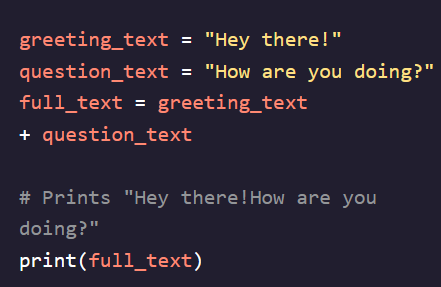
**Literal Numbers:**

**-** Actual numbers that are hardcoded into the script, not just a variable name (number 3 in the above example)

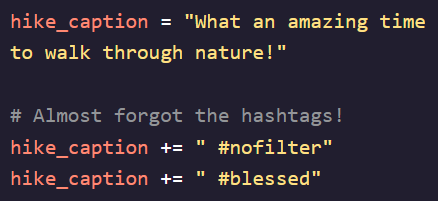
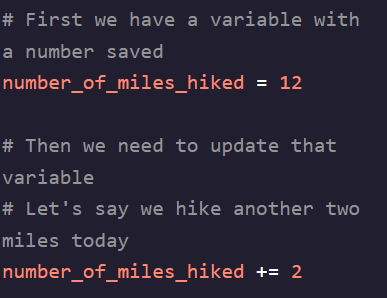
**Arithmetic Operations:**

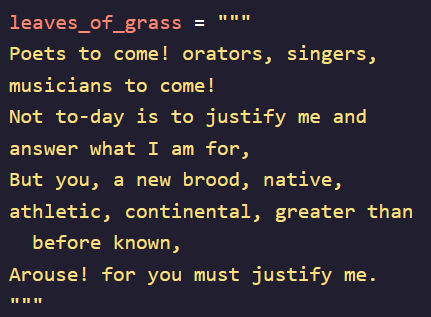
- **Addition** = +  
- **Subtraction** = -  
- **Multiplication** = \*  
- **Division** = / (Python converts all ints to floats before dividing)   
- **Exponents** = \*\* (Allows you to put any number after to raise a number to that power – 5\*\*3 = 125)  
- **Modulo** = % (Returns either 0 or remainder of two numbers being divided = 4%2 = 0 but 5%2 = 1)   
 

**String Concatenation:**

- Used to combine two strings into a new one that is a combination of both of them   
- Can add space between concatenations using “ “   


**Plus-Equals (+=):**

- A shorthand way to update variables by adding the current value of another number to it  
- Can also be used for string concatenation  


**Multi-Line Strings (” ” ”):**  
**-** Used to create strings that are multiple lines instead of just one  


**User Input:**

**-** When we need a user to assign values to variables we can use the **input()** function  
- This can be dangerous, however, as users who enter incorrect data or data in the wrong format can break your script  
- Necessary to put user input into a Try/Except block or integrate some sort of input validation testing   
